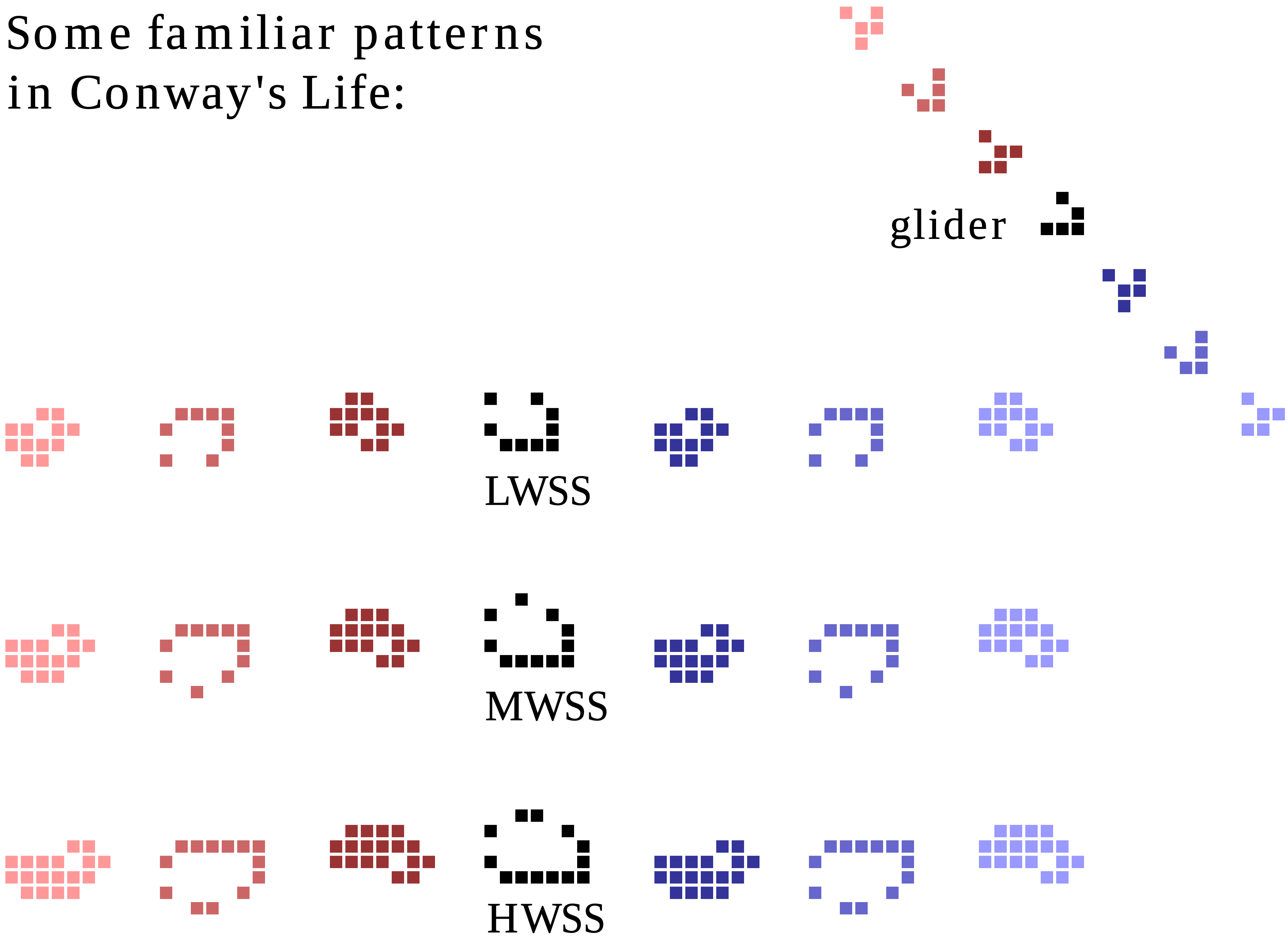
Searching for Spaceships

David Eppstein
UC Irvine, Dept. Inf. & Comp. Sci.

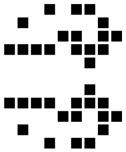
http://arXiv.org/abs/cs.AI/0004003
http://www.ics.uci.edu/∼eppstein/ca/

MSRI Combinatorial Game Theory
Research Workshop, July 2000

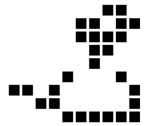
Some familiar patterns in Conway's Life:



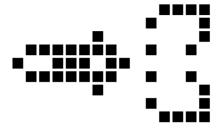
...and some less familiar patterns...



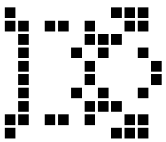
2c/4



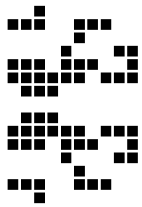
8c/16



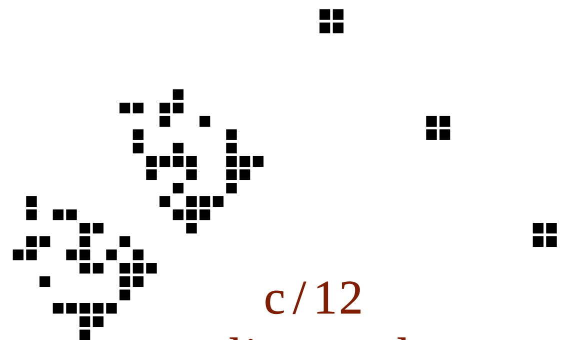
6c/12



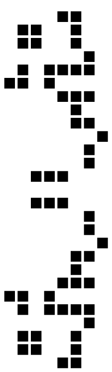
c/3



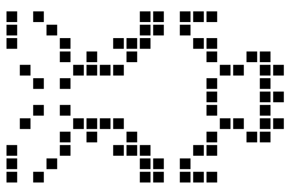
2c/5



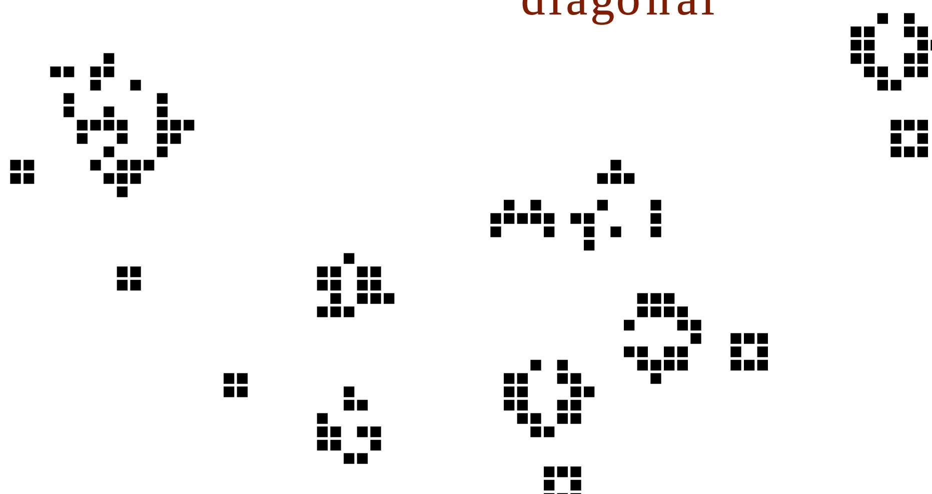
c/12
diagonal



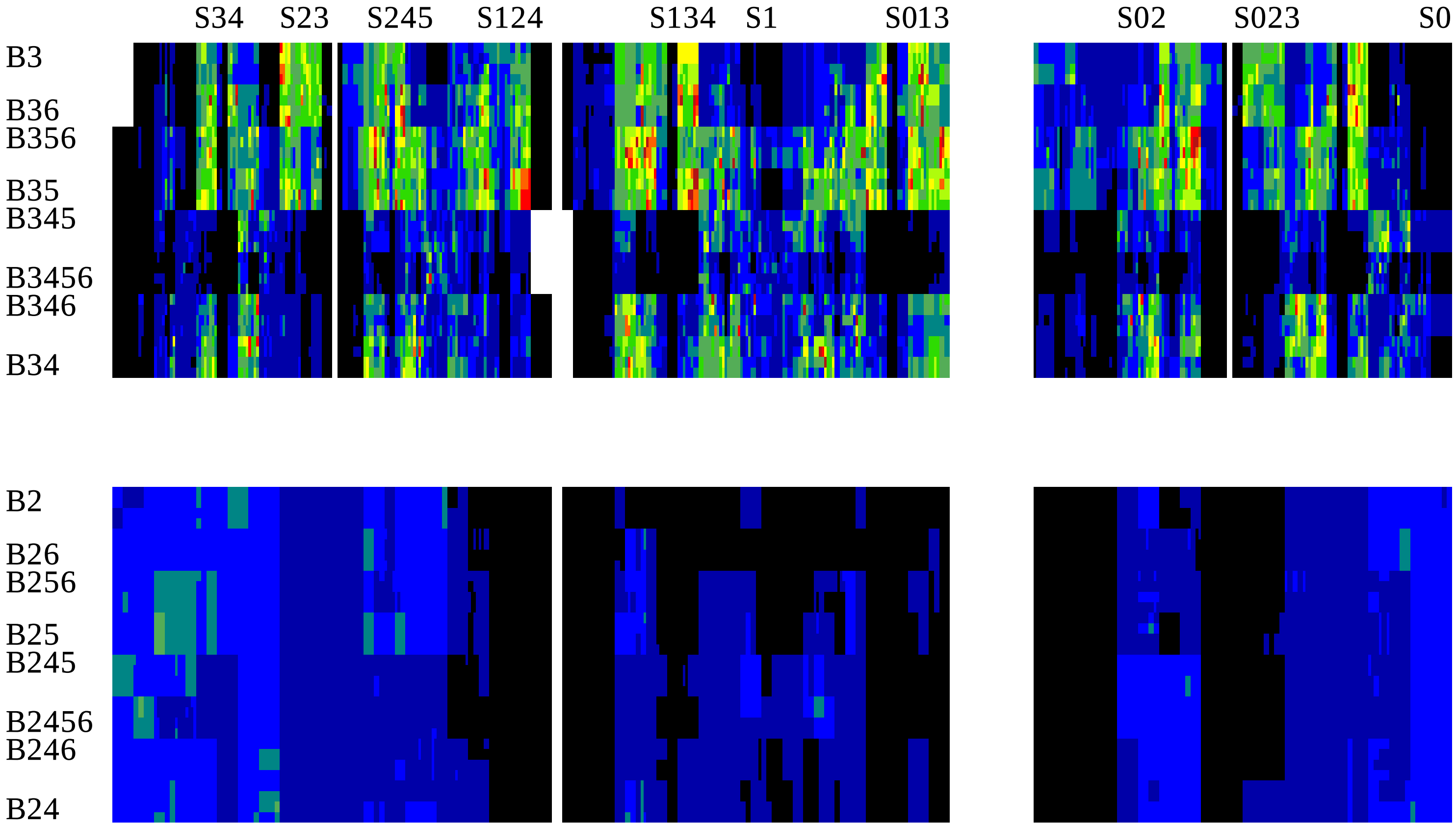
c/5



c/4



Which semitotalistic rules support spaceships?



Key: color = number of spaceship velocities known

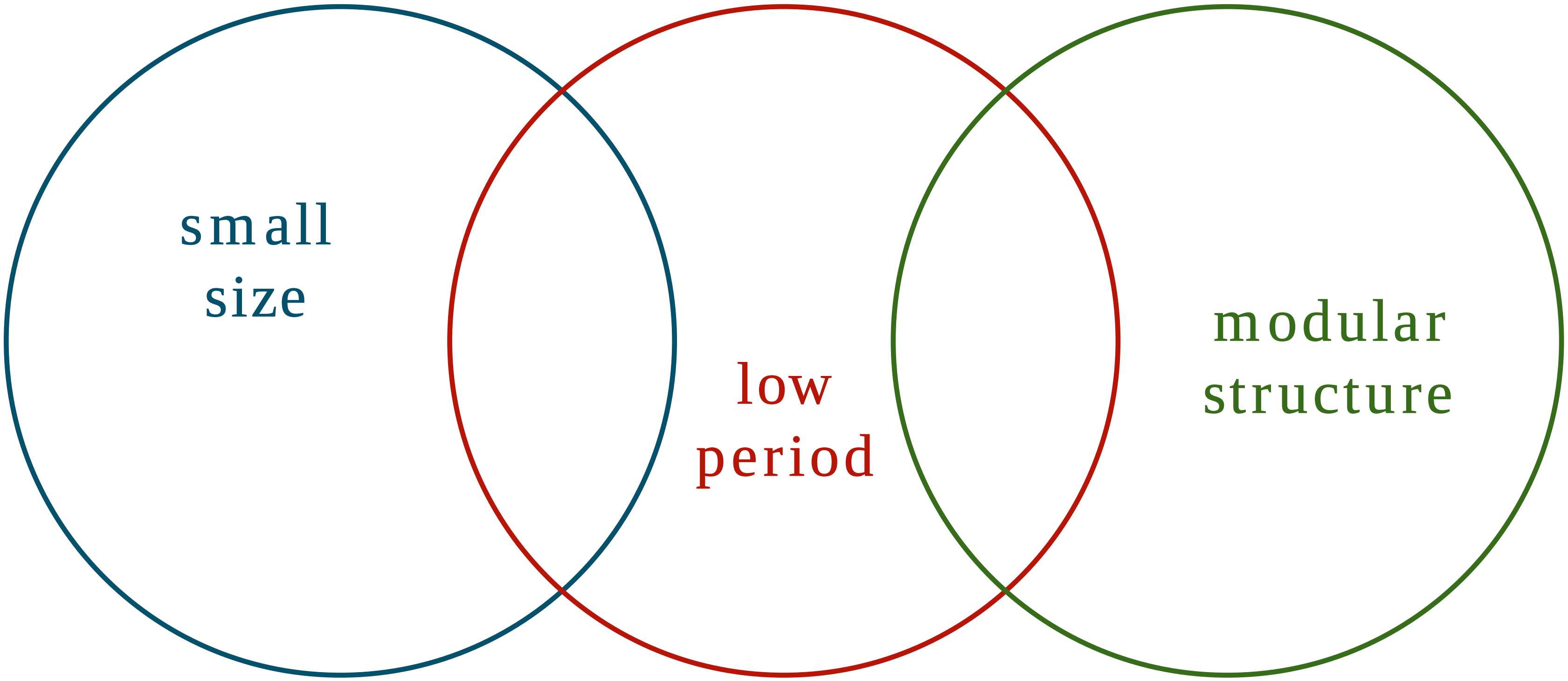
= 0 = 1 = 2 = 3 = 4 = 5

= 6 = 7 = 8 = 9 > 9 impossible

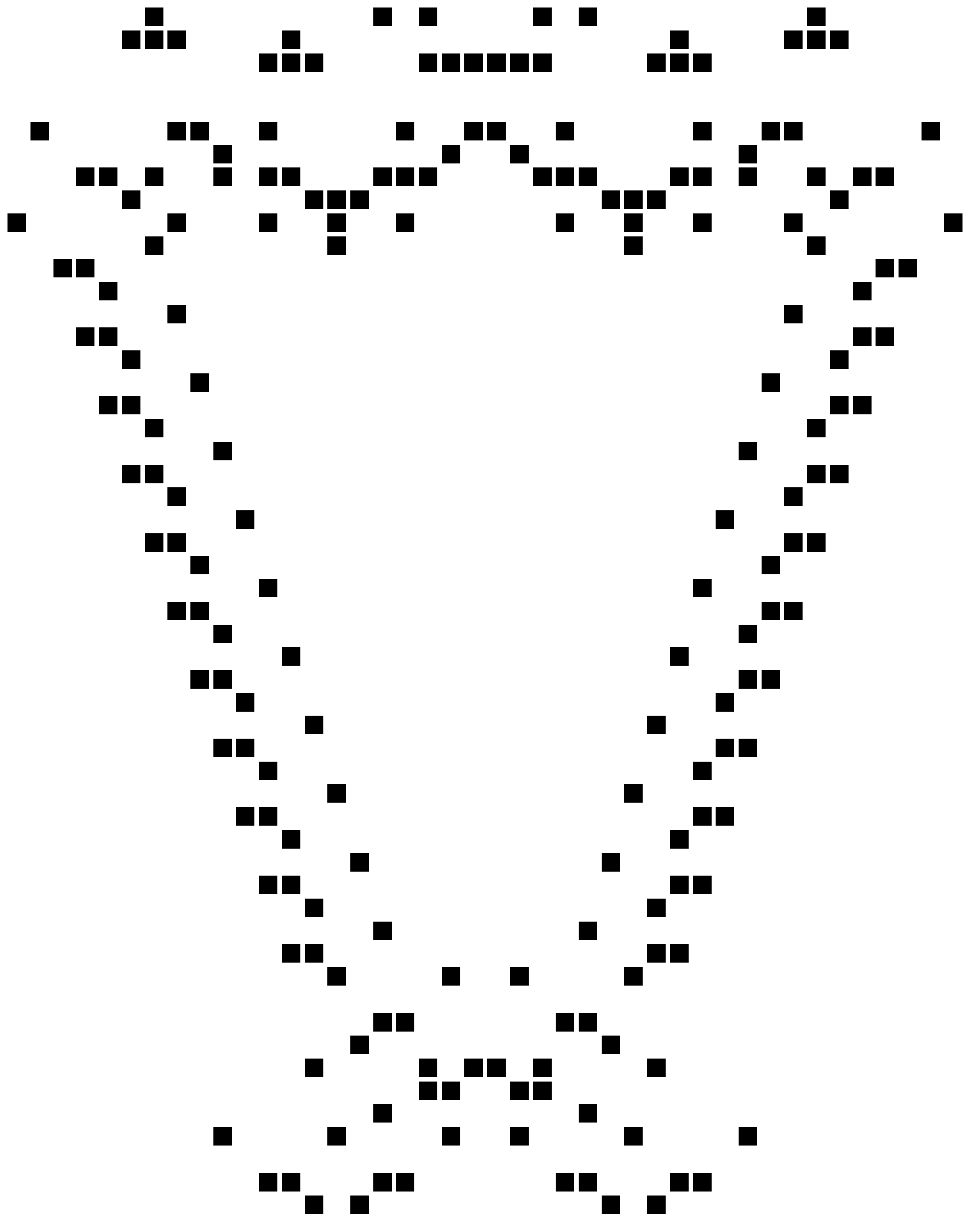
Goals:

- Understand spaceship structure
- Determine which rules support spaceships
- Determine which rules support
  which spaceship velocities
- Identify potentially interesting rules
- Discover new spaceships in Life
  and other interesting rules
- Categorize new types of pattern

Categories of spaceship
(Venn diagram):



A modular low-period spaceship?



c/2 spaceship in rule B27/S0

Previous spaceship searchers

Various brute force programs
for all/random small patterns. . .

lifesrc (Hickerson/Bell)

Depth-first search maintains the state of each phase of each cell as unknown, live, dead, or don't care; recursively tries setting each unknown cell to live or dead and propagating the consequences to neighbors.

knight (Coe)

Breadth-first search extends partial patterns a row at a time in one phase only; checks that evolving the pattern through a period results in the same rows. Uses a fixed amount of depth-first lookahead to limit queue size.

General Design Ideas

Fast programs come from fast algorithms

Dynamic programming:

> Remember previously solved subproblems
>
> Before solving a new subproblem
> test if equivalent one already seen
> and if so look up solution

What are the subproblems?

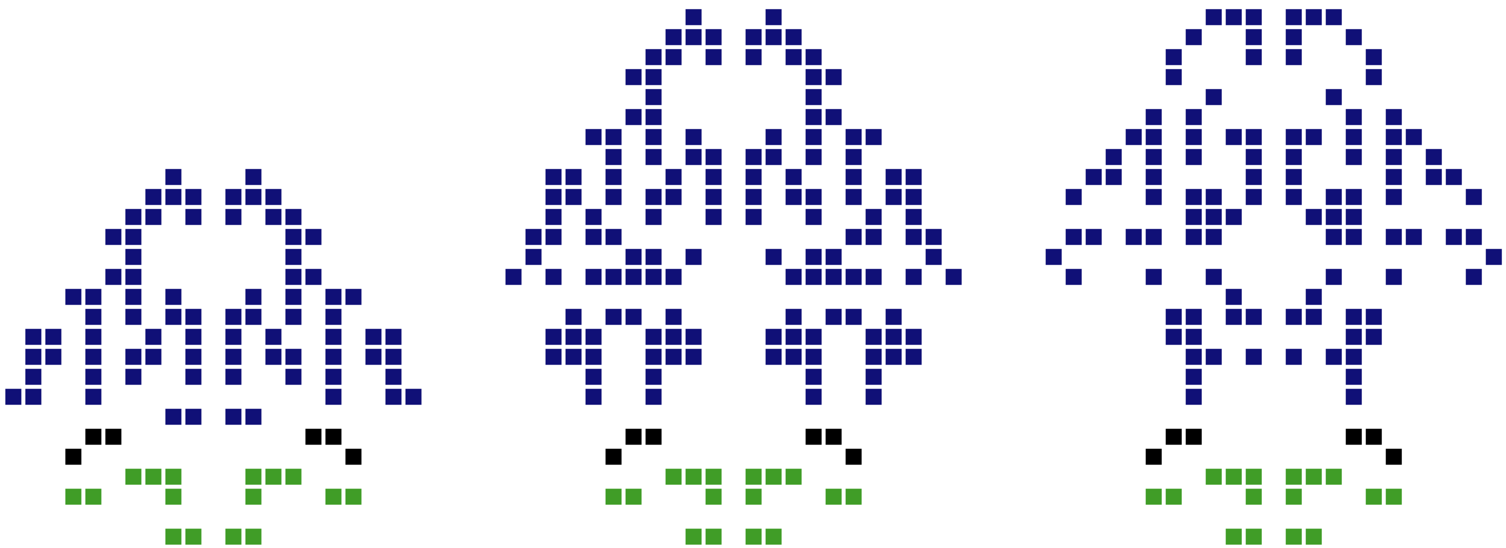
- Partial patterns consisting of cells in each phase of each row from front of spaceship back to some cutoff
- Solution = can this partial pattern be completed to form a spaceship?

When are subproblems equivalent?

- When the last two rows in each phase are equal
- Can cut and paste any solution of one to solve the other

Example of equivalent partial patterns

(c/2 Life spaceships with double domino tailspark)



The Search Algorithm

For each reachable state
(equivalence class of partial patterns)

if complete pattern, output and terminate

else for each successor
(formed by adding one row in one phase
with correct evolution of previous rows)

add to list of reachable states

Analysis:

$\mathcal{O}(2^{2pw})$ possible states
$\mathcal{O}(2^{w})$ possible successors/state
$\mathcal{O}(2^{(2p+1)w})$ total time

Details

To remember which states already seen:

- Use hash table
- May give false negative (hash collision)
  but never false positive

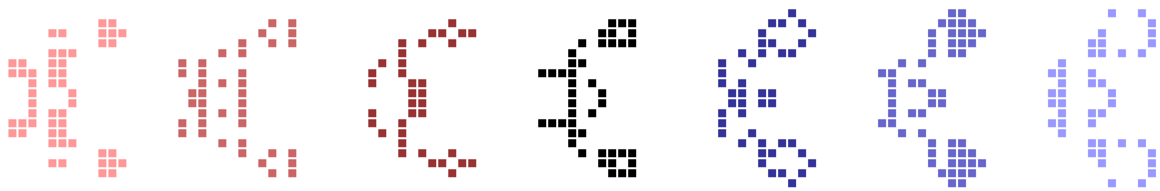
To find successors of a state:

- Represent as paths in a graph
- Fast bit-parallel path-listing algorithm
- Incorporate lookahead

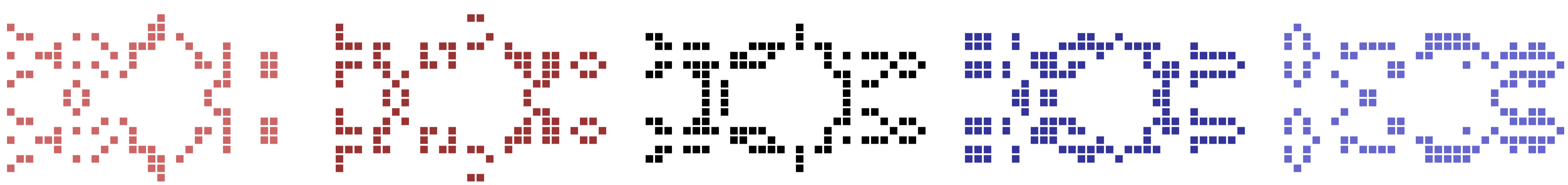
Overall search structure:

- Hybrid algorithm combining breadth-first
  and iterative deepening techniques

New spaceships in Life (B3/S23):

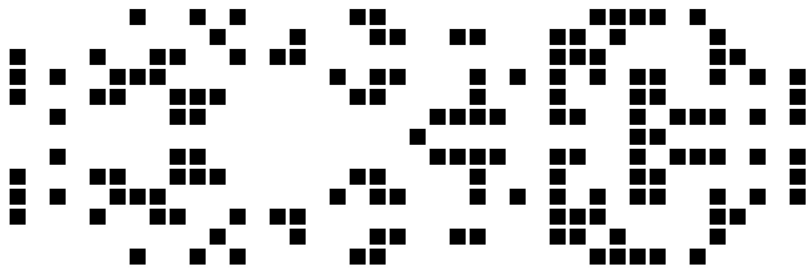


2c/7 “Weekender” [DE]

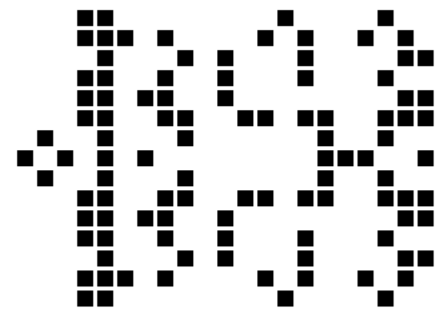


c/6 “Dragon” [Paul Tooke]

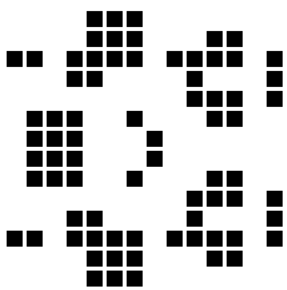
New spaceships in HighLife (B36/S23):



c/4



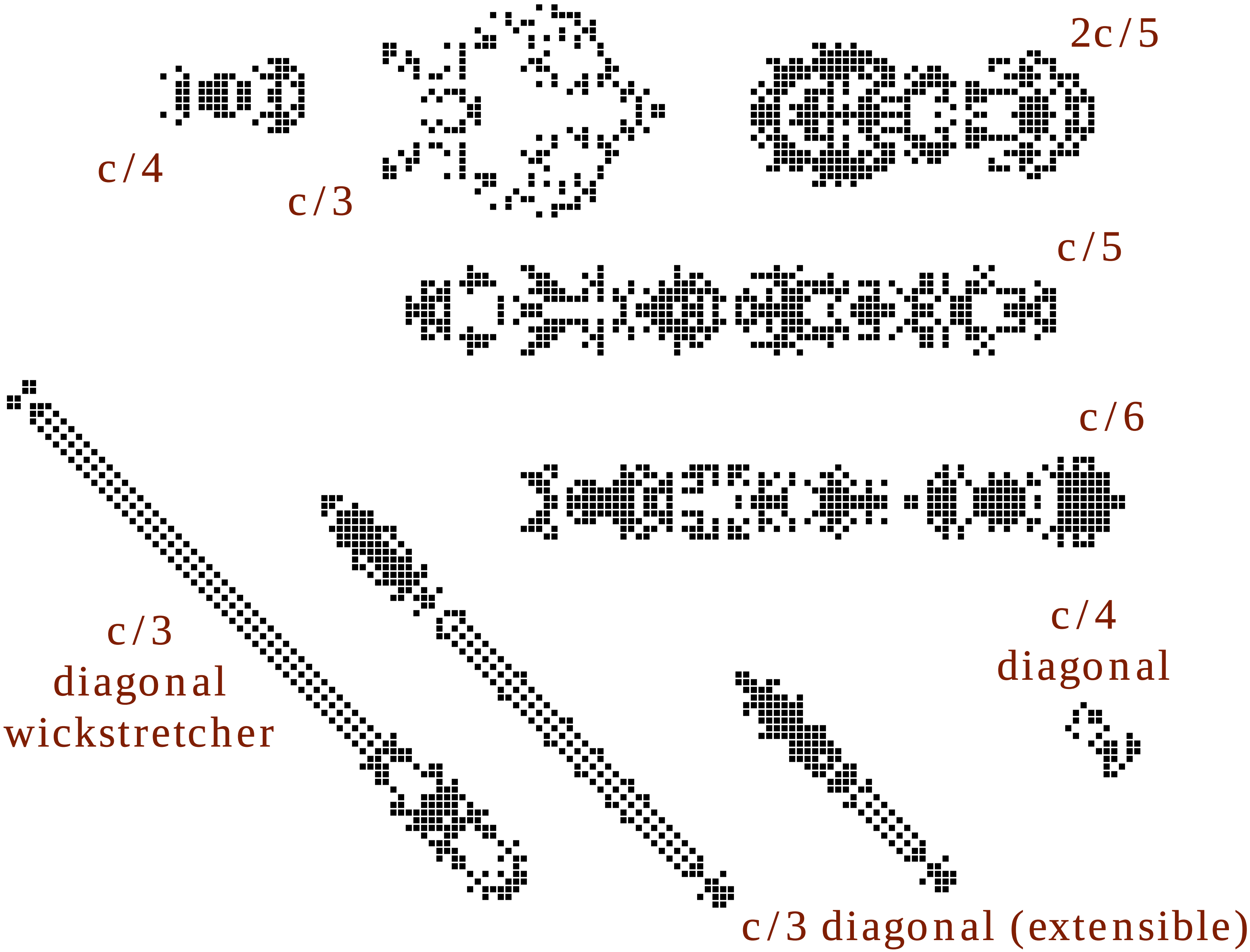
c/3



c/5

2c/5

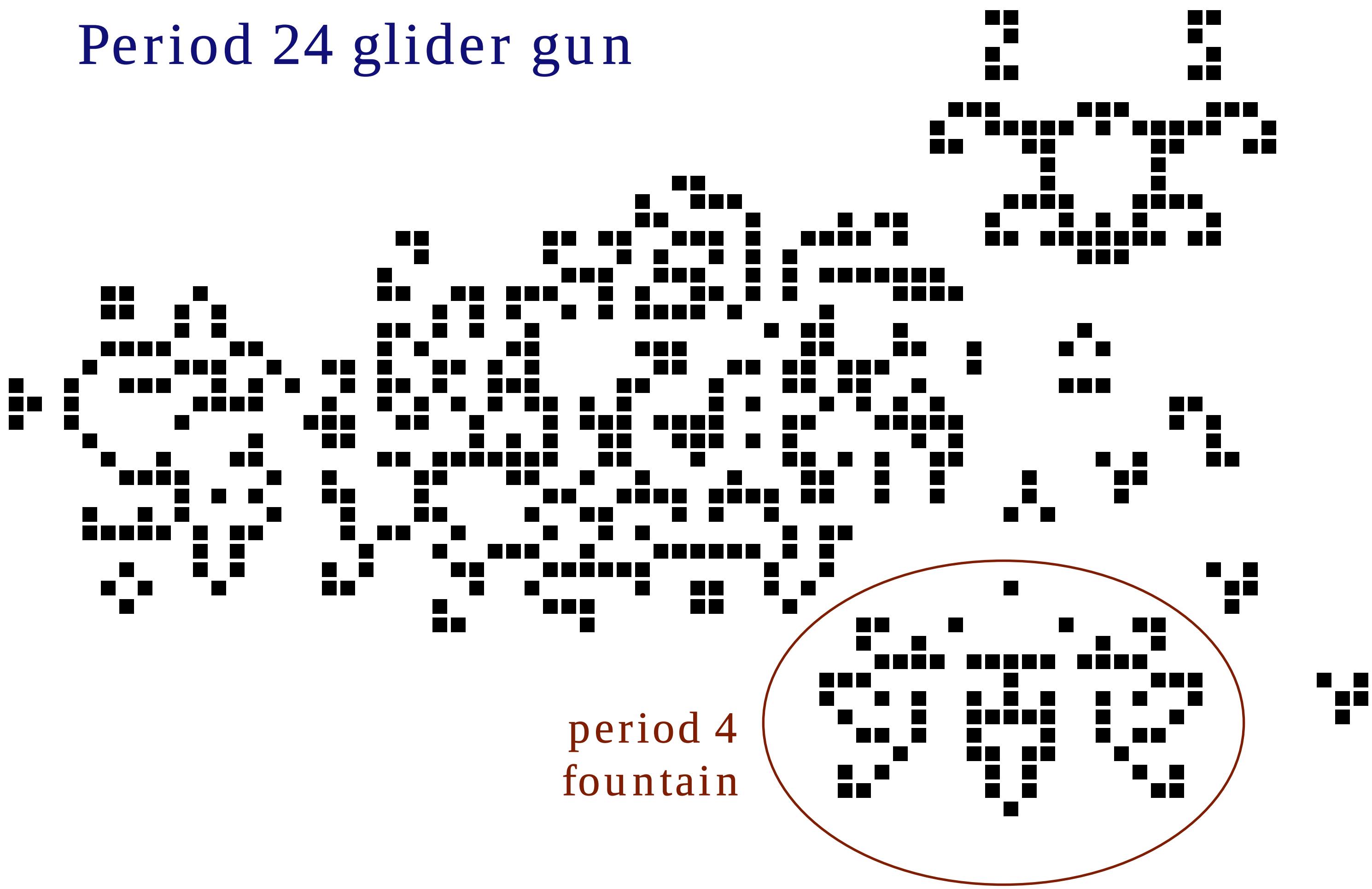
New spaceships in Day & Night (B3678/S34678)



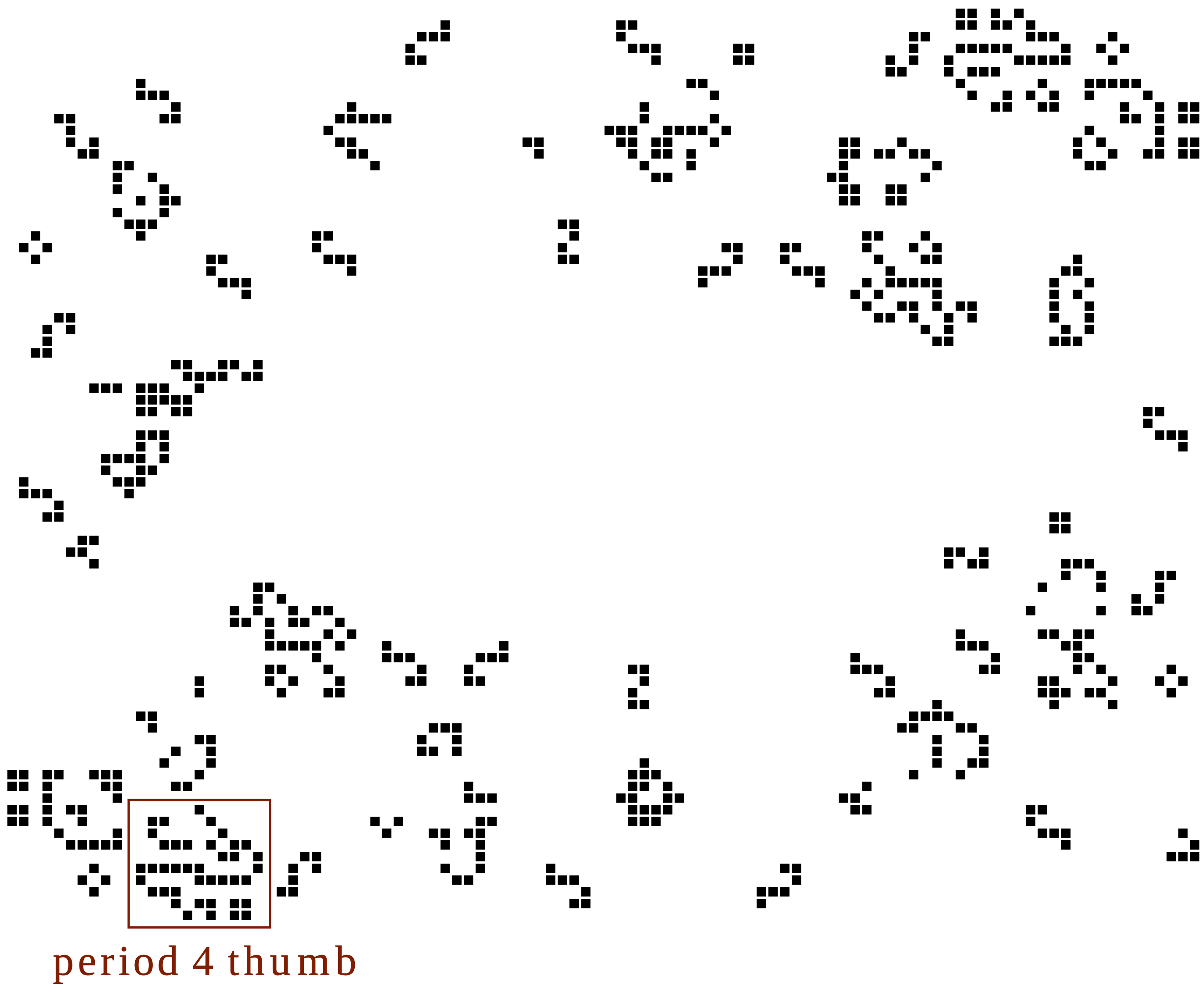
Recent work:

- Adapt algorithms to oscillator searches
  - Three layers: list rows extending a single phase, find consistent set of extensions from each phase, global search
  - Special role for stators and sparks
  - Smart termination detection allows asymmetric stator to finish symmetric oscillator
  - Adapt KMP string matching to test whether finished pattern has desired period
- Use new oscillators to reduce size of Life guns
- Direct search for low-period guns

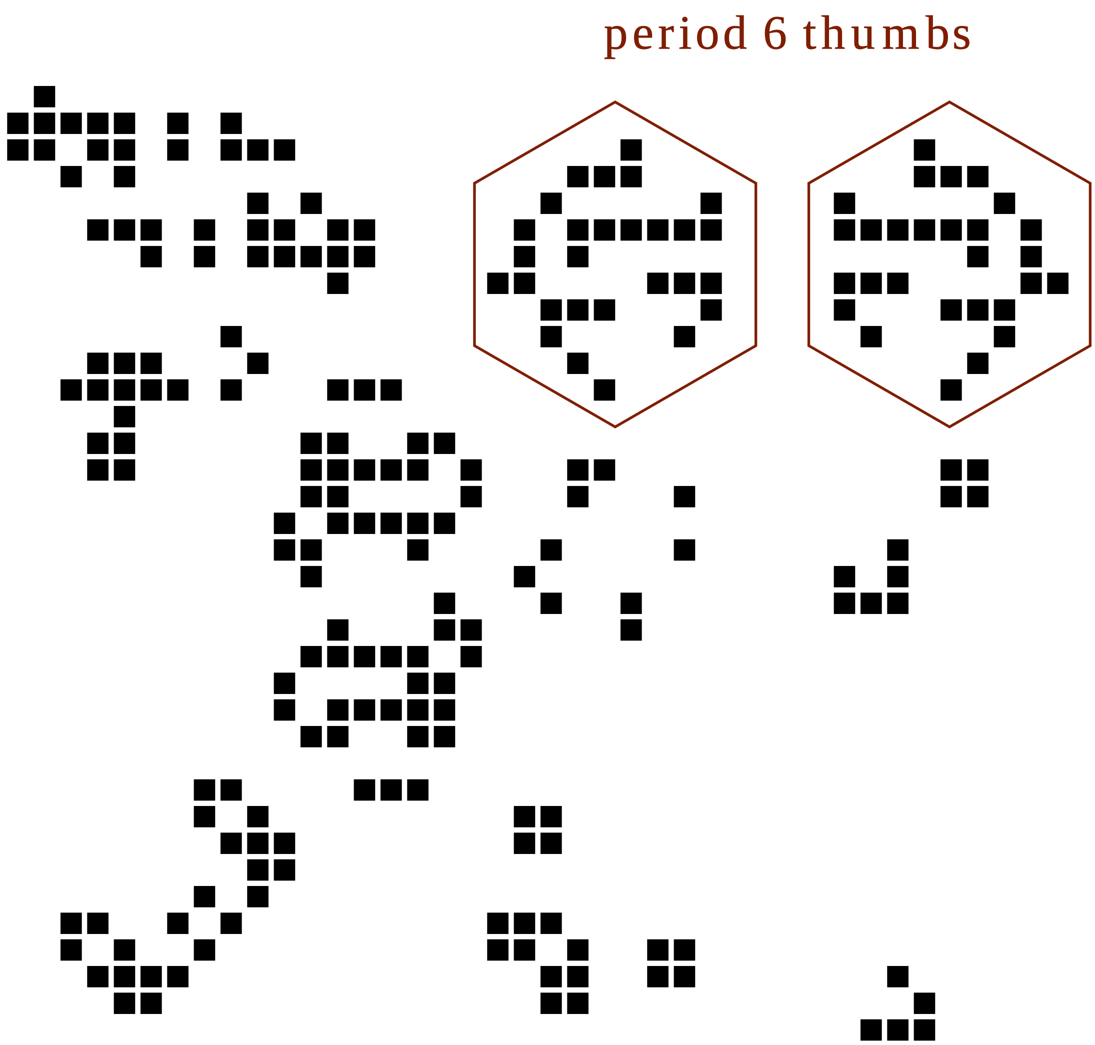
Period 24 glider gun



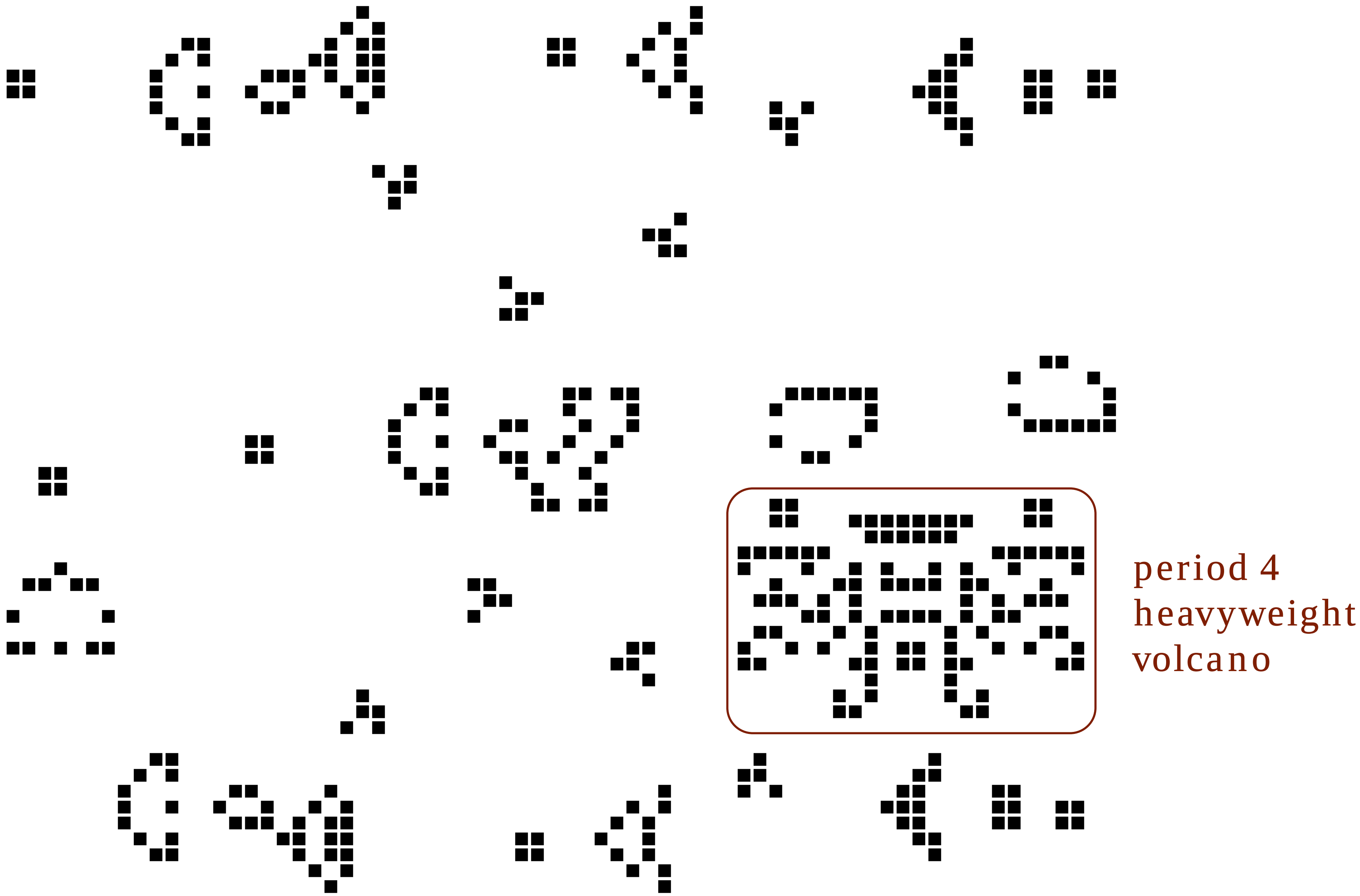
Period 68 glider gun

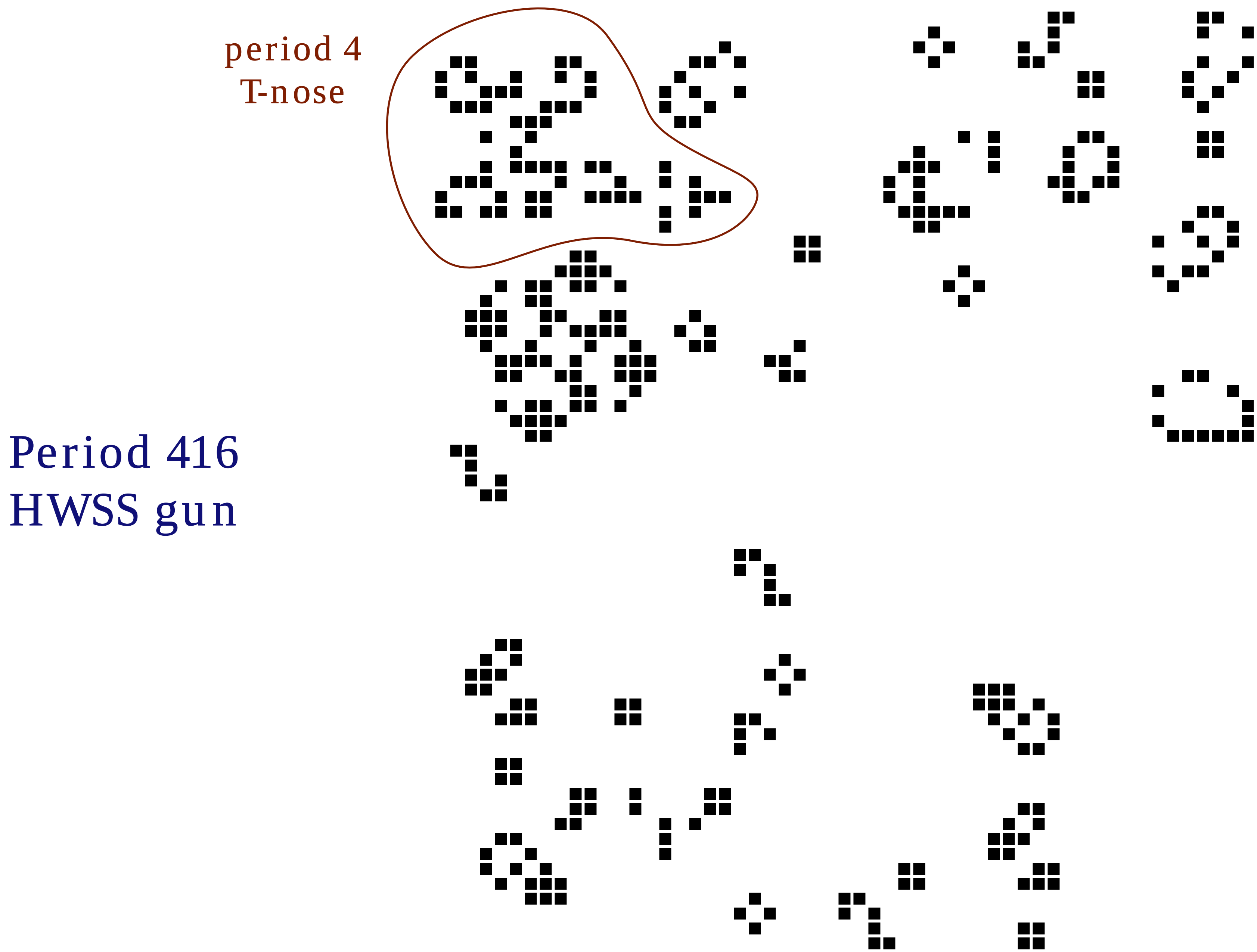


Period 72 glider gun



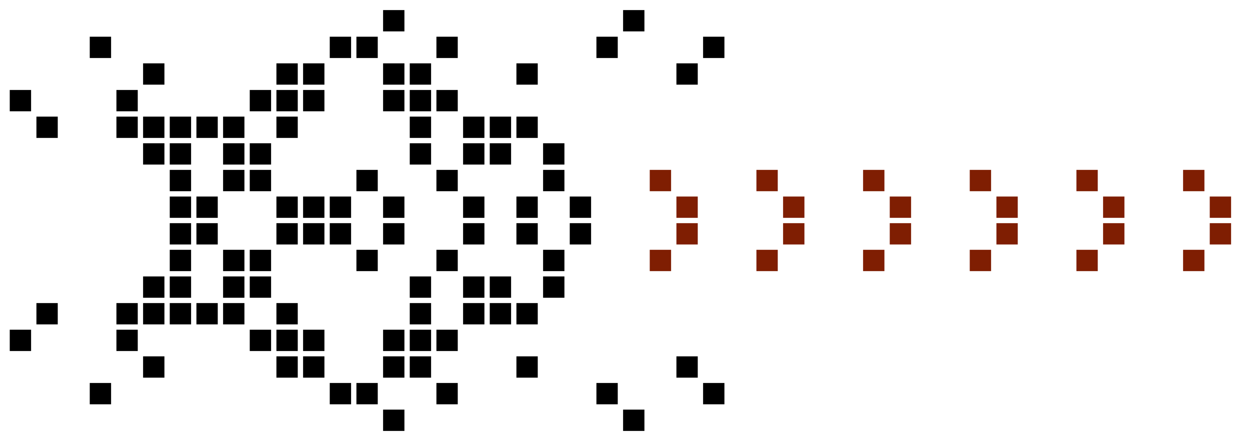
Period 60 HWSS gun





period 4
T-nose
Period 416
HWSS gun

Period 4 ray gun in B25/S45



Conclusions

- Spaceships are ubiquitous
- Life good but not outstanding in terms of its variety of spaceships

  more important: simple glider-generating reactions
- CA behavior on random conditions does not predict which non-random patterns exist
- Exponential algorithms can still be effective

A question of complexity

Is the following problem decidable?

> INPUT: A semi-totalistic (Moore neighborhood) rule and a velocity
>
> OUTPUT: A spaceship with that velocity, or "impossible" if none exists

I.e. is there an effective bound for the size needed to achieve a given velocity?

All the complexity has to be in the velocity, since there are only $\mathcal{O}(1)$ possible rules